1. State Goals and Define the System

CPU: Intel® Xeon® Silver 4314 CPU @ 2.40GHz

Cores: 16

Network Interface Card: Mellanox Technologies MT2892 Family [ConnectX-6 Dx]

RAM: 131356700 kB Total (=131.3567 GB)

OS: Debian GNU/Linux 11

Kernel: 5.10.0-28-amd64

Goals of the study:

Determine how the implemented socket architecture for inter-process communication performs compared to the previously developed shared memory architecture by Paul Raatschen. The comparison should be fair and measure performance under identical scenarios. Finally, make a verdict on which architecture is more performant.

2. List Services and Outcomes

Replicate previous experiment by Paul Raatschen.

Both implementations provide the same service to the end-user. They receive a steady stream of wanted communication request and a significantly larger stream of unwanted request (simulating a DoS attack).

Possible outcomes are that the software is either able to reliably parse log messages to the intrusion prevention system (simplefail2ban) or that it will become overwhelmed by the incoming communication requests.

3. Select Metrics:

Choosing similar performance metrics according to the previous experiments by Paul Raatschen is vital to make direct comparisons possible.

Performance metrics (directly quoted from Paul Raatschens bachelor thesis):

- Total amount of unwanted requests dropped (number of packets)
- Total amount of unwanted requests dropped, relative to the total amount of unwanted request sent (percentage)
- Amount of log messages processed by Simplefail2ban, relative the amount of log messages sent by the test server (percentage)
- Central Processing Unit (CPU) utilization of Simplefail2ban (seconds of CPU time)

4. List Parameters

Again, following in the footsteps, the parameters affecting performance will overlap heavily with the experiments performed by Paul Raatschen (directly quoting P.R. bachelor thesis):

- CPU: 16 cores; no hyper-threading enabled

- NIC: MTU 1500 bytes
- Trex: One interface, 30 threads *(this might change)*
- Number of entries in eBPF maps for IPv4 & IPv6: 1000000
- Number of receiving threads used by the test server: 16 *(this might change)*
- Duration of measurement: 300 seconds *(this might change)*
- Amount of valid traffic sent: 10000 PPS *(this might change)*
- Number of clients sending valid traffic: 254 *(this might change)*
- Simplefail2ban parameters:
    - Number of banning threads used: 4 (shared memory and sockets), 1 (logfile)
    - Number of hash table bins used: 6000011 *(Why exactly this size? I get that it is supposed to be a prime number, but why not any other arbitrary prime number instead?)*
    - Ban threshold for clients: 3
    - Ban time for clients: 30 Seconds *(this might change)*
    - Line count for the shared memory buffer segments: 100000
    - Segment count for the shared memory buffer: 16
    - Buffer size for uring_getlines: 2048
    - Number of sockets: Same amount as the number of readers (= 1) *(This seems like an obvious bottleneck)*

5. Select Factors to Study

- Effects of differing amount of invalid traffic sent: 100k, 1m, 10m PPS
- Effects of differing number of clients sending invalid data: 65534, 131086
- Effect of using socket architecture

6. Select Evaluation Technique

Measuring a system in which the traffic that is received is simulated

7. Select Workload

Steady stream of both valid and invalid traffic from several clients

8. Design Experiments

Work in Progress

9. Analyze and Interpret Data

Not yet possible

10. Present Results

Not yet possible